似然函数学习笔记

战义强
Email: zhanyq8@mail.sysu.edu.cn

最后更新：2024 年 11 月

此学习资料是关于正态分布、二项分布等似然函数以及 R 语言实现，供课题组研究生学习使用。学习本材料之前，需要了解什么是概率密度函数 pdf、什么是累积概率密度函数 cdf。

1 什么是似然函数

在数理统计学中，似然函数（英语：likelihood function）是一种关于统计模型中的参数的函数，表示模型参数中的似然性（英语：likelihood）。似然函数在统计推断中有重大作用，如在最大似然估计和 Fisher Information 之中的应用等等。文字意义上，“似然性”与“或然性”或“概率”意思相近，都是指某种事件发生的可能性，但是在统计学中，“似然性”和“概率”（或然性）有明确的区分：概率，用于在已知一些参数的情况下，预测接下来在观测上所得到的结果；似然性，则是用于在已知某些观测所得到的结果时，对有关事物之性质的参数进行估值，也就是说已观察到某事件后，对相关参数进行猜测.

2 正态分布的概率密度函数、似然函数

正态分布的似然函数用于估计正态分布的参数, 通常是均值 μ 和方差 σ^2, 假设我们有一个样本数据集 $(x_1, x_2, ..., x_n)$, 其中来自正态分布，其概率密度函数 (probability density function, pdf) 为：

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{1}$$

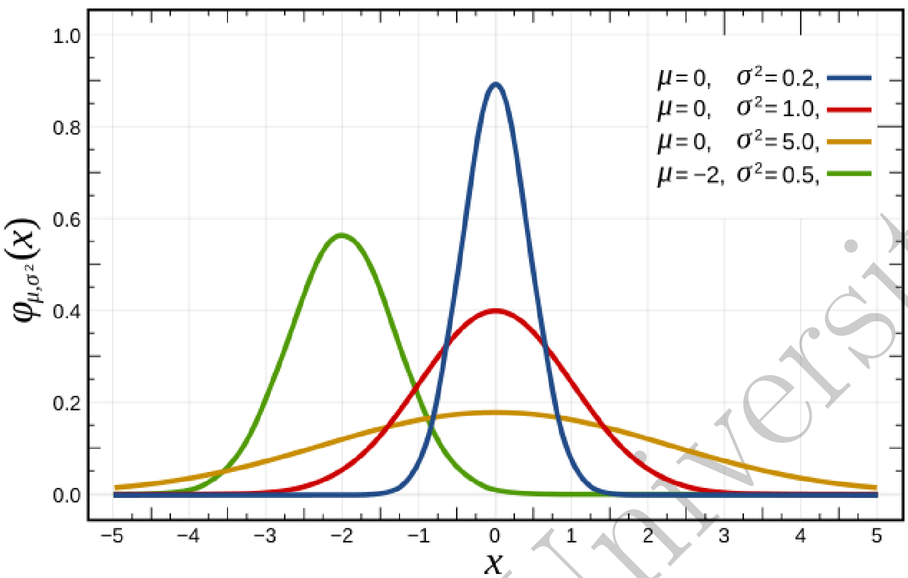

图 1: 概率密度函数图

图 1 展示了不同均数和标准差条件下，概率密度函数的分布图。在标准正态分布中，随机变量 X 的取值范围是 $(-\infty,\infty)$, $\mu=0$, $\sigma=1$, 当 $X=0$ 时，$f(x=0|\mu=0,\sigma=1)=\frac{1}{\sqrt{2\pi}}\approx 0.3989$. 在标准正态分布的概率密度图上 $X=0$ 对应的值大约是 0.3989。这个值的意义在于，它代表了标准正态分布中随机变量取值为 0 的概率密度。在正态分布中，概率密度函数描述了随机变量取特定值的可能性，而不是取特定值的概率（因为连续随机变量取任何具体值的概率都是 0）。概率密度函数在某个点的值越高，随机变量落在该点附近的概率就越大。

似然函数 $L(\mu,\sigma^2)$ 是随机变量 X 的样本的概率密度的乘积：

$$L(\mu,\sigma^2)=\prod_{i=1}^{n}\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right) \tag{2}$$

通过取对数，进一步可以写为加法的形式：

$$\log L(\mu,\sigma^2)=-\frac{n}{2}\log(2\pi\sigma^2)-\frac{\sum_{i=1}^{n}(x_i-\mu)^2}{2\sigma^2} \tag{3}$$

假设现在有一组样本 (5.1, 4.9, 6.2, 5.7) 来自正态分布分布，我们要计算出该样本所在的总体的均数和标准差，如何使用似然函数进行计算呢？该

样本的似然函数如下：

$L(\mu, \sigma^2) = f(x = 5.1|\mu, \sigma^2) f(x = 4.9|\mu, \sigma^2) f(x = 6.2|\mu, \sigma^2) f(x = 5.7|\mu, \sigma^2)$

我们需要找到不同的 μ, σ 组合，使 $L(\mu, \sigma^2)$ 的值最大。当然，L 是单调递增函数，如果 $L(\mu, \sigma^2)$ 的值最大, $\log L$ 也是最大，并且 $\log L$ 的计算方法是加法，软件更容易计算。这个探索的过程，由软件来帮我们自动实现。

下面，我们使用一个简单的例子，一个 10000 个观测值的样本 x，来自于均数是 3、标准差是 2 的总体，我们使用 R 语言通过样本来估计总体的均数和标准差：

```r
# 定义似然函数
loglikelihood1 <- function(x, par) {
    mu = par[1]
    sigma = par[2]
    n = length(x)
    loglike = -n/2 * log(2 * pi * sigma^2) - sum((x - mu)^2) / (2 * sigma^2)
    return(-loglike)
    # 这里加了个负号，因为R中的optim函数默认是进行minimization计算，而不是maximization
}

#或者直接使用dnorm函数
loglikelihood2 <- function(x, par) {
    mu = par[1]
    sigma = par[2]
    n = length(x)
    loglike = sum(dnorm(x, mu, sigma, log = TRUE))
    return(-loglike)
}

# 示例数据
set.seed(1234);x <- rnorm(10000, mean = 3, sd = 2)

# 使用MASS包的optim函数来寻找最大似然估计
initialvalues = c(mean(x), sd(x))
mle <- optim(par = initialvalues, fn = loglikelihood1, x=x)
#或者
mle <- optim(par = initialvalues, fn = loglikelihood2, x=x)

mle_mu <- mle$par[1];
mle_sigma <- mle$par[2]

cat("MLE for mean:", mle_mu, "\n")
cat("MLE for standard deviation:", mle_sigma, "\n")
```

下面，我们再进一步，展示一个双变量的例子, 假设样本中有另外一个变量 y，其与 x 的关系是 $\hat{y} = \beta_0 + \beta_1 x$，我们使用最大似然估计来计算中的 β_0 和 β_1.

在这个例子中，变量 y 的概率密度函数是：

$$f(y \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right) \tag{4}$$

其中 $\mu = \beta_0 + \beta_1 x$

```r
# 定义似然函数
loglikelihood3 <- function(data, par) {
    x = data[,1]
    y = data[,2]
    mu = par[1] + par[2] * x
    sigma = par[3]
    n = length(x)
    loglike = -n/2 * log(2 * pi * sigma^2) - sum((y - mu)^2) / (2 * sigma^2)
    return(-loglike)
}

loglikelihood4 <- function(data, par) {
    x = data[,1]
    y = data[,2]
    mu = par[1] + par[2] * x
    sigma = par[3]
    n = length(x)
    loglike = sum(dnorm(y, mu, sigma, log = TRUE))
    return(-loglike)
}

# 示例数据
set.seed(1234); nobs < 10000
x <- rnorm(nobs, mean = 3, sd = 3)
beta0 <- 2; beta1 <- 0.5
y <- beta0 + beta1 * x + rnorm(n)
data <- data.frame(x=x, y=y)

# 使用MASS包的optim函数来寻找最大似然估计
mle <- optim(par = c(2,0.5,2), fn = loglikelihood3, data=data); mle
mle <- optim(par = c(2,0.5,2), fn = loglikelihood4, data=data); mle

beta0 <- mle$par[1]
beta1 <- mle$par[2]
```

3 二项分布的概率密度函数、似然函数

未完待续。

4 Cox 模型的偏 MLE 估计